

## print

Мы уже знаем, что выводить строки можно с помощью команды `print()`. Есть еще похожая команда `println()`. Она не только выводит строку, но и переводит после этого курсор в консоли **на новую строку**.

Для лучшего понимания рассмотрим два примера.

```
fun main() {  
    print("Hello") // вывод  
    print("World!") // вывод  
}
```

language-kotlin

Получим строку:

```
HelloWorld!
```

language-undefined

Дело в том, что после команды

```
print("Hello");
```

language-kotlin

курсор в консоли остается на той же строке. Туда же вторая команда дописывает строку **"World!"**. Таким образом и получается склеенная строка.



А если используем `println()`:

```
fun main() {  
    println("Hello") // вывод и перенос на новую строку  
    println("World!") // вывод и перенос на новую строку  
}
```

language-kotlin

Вывод будет следующим:

```
Hello  
World!
```

Дело в том, что после команды

```
println("Hello")
```

language-kotlin

курсор передвинулся на следующую строку. И уже на новой строке выводится "World!".

## Вывод информации

С помощью этих команд можно выводить **числа** или результат вычисления выражений. Например:

```
fun main() {  
    println(26)  
  
    println(1024 + 48) // 1072  
  
    println(5 + 8 * 2) // 21  
  
    print((5 + 8) * 2) // 26  
}
```

language-kotlin

Через `//` указано то, что будет после исполнения команды. То есть вывод будет следующим:

```
26  
1072  
21  
26
```

language-yaml

Как вы могли заметить, каждый результат выводится на каждой строке. При этом **операции** над числами внутри команды осуществляются по правилам **математики**: числа складываются, умножаются и так далее.

## Форматированный вывод

Довольно часто есть необходимость выводить числа и строки вместе. Просто вывод чисел для пользователя ничего не означает. Иногда надо бывает перед или после числа добавить объяснения.

Есть 2 способа добиться этого результата:

- с помощью конкатенации (склеивания строк)
- с помощью знака доллара

В этом разделе рассмотрим первый способ, а второй рассмотрим чуть позже.

## Конкатенация

Итак, нам нужно объединить строку с числом. Для этого строку нужно обозначить в кавычках (чтоб компилятор ее "узнал" как строку), а между строкой и числом нужно поставить знак плюс (+), который тут просто приклеивает их. Результатом такой склейки получается **СТРОКА**.

Для лучшего понимания рассмотрим код:

```
fun main() {                                     language-kotlin
    println("5 + 7 = " + 12) // 5 + 7 = 12

    println("5" + "7 = 57") // 57 = 57

    println("Количество месяцев в году: " + 12)

}
```

Результатом работы данной программы будет:

```
5 + 7 = 12                                     language-undefined
57 = 57
Количество месяцев в году: 12
```

Но знак плюс мы можем поставить между строкой и числом только в том случае, если первым слагаемым является строка, а вторым — число. Для понимания рассмотрим следующий пример с кодом, который выведет **ошибку**.

```
fun main() {                                     language-kotlin
    println(2022 + " год будет лучше 2021-го")    // ✗ - Ошибка!! Нельзя
    println("Самый лучший год - " + 2011)        // ✓

    println("Kotlin появился в " + 2011 + " году") // ✓
}
```

Первая строка ( `2022 + " год будет лучше 2021-го"` ) которую мы пытаемся вывести написана с ошибкой - сначала записано число, и к этому числу мы пытаемся присоединить строку. Так **Kotlin** делать не позволяет.

Вторая строка ( `"Самый лучший год - " + 2011` ) записана правильно - к строке прибавляем (приклеиваем) число.

И третья строка ( `"Kotlin появился в " + 2011 + " году"` ) тоже записана правильно. Сначала к строке прибавляем (приклеиваем) число — получаем **строку**. И вторым плюсом приклеиваем конец строки.

Заметьте, что пробел ( `" "` ) – это тоже строка, причем **не пустая**. Пример показывает, что один и тот же результат вывода можно достичь **разными способами**.

### Примечания:

1. То, что мы пишем в круглых скобках у команд `print()` и `println()` называется **аргументами** или **параметрами** команды.
2. Все команды, которые мы уже рассмотрели и рассмотрим будущем, записываются в приведенном в лекциях формате, другое написание недопустимо, так как в `Kotlin` строчные и заглавные буквы различны.
3. Команда `println()` с пустым списком аргументов (пустые круглые скобки) просто вставляет новую пустую строку. Например:

```
println("Строка 1");  
println();  
println("Строка 3");
```

language-kotlin

выведет на экран три строки, одна из которых пустая:

Строка 1

Строка 3